WHAT IS CLAIMED IS:

1. A method for managing a bad block in a flash memory, comprising:

(a) allocating a used area comprising a plurality of used blocks and a spare area comprising a plurality of spare blocks in the flash memory, and providing a block map page group including a plurality of block map pages in which mapping information to map a bad block generated in either the used area or the spare area to a first spare block is disposed;

(b) having the mapping information of the block map page selected by a predetermined rule, reside in the block map page group in the flash memory as resident mapping information; and

(c) mapping the bad block generated during a flash operation to an unused spare block found through the resident mapping information, updating the resident mapping information as updated mapping information, and recording the updated mapping information in the block map page belonging to the block map page.

2. The method as claimed in claim 1, wherein step (a) comprises:

(a1) allocating the used area comprising the plurality of used blocks and the spare area comprising the plurality of spare blocks in the flash memory;

(a2) generating block map fields on each of the spare blocks, on which mapping information between the generated spare blocks and the bad blocks is

to be generated as generated block map fields, and initializing block map information containing the generated block map fields, a count field for determining a time of the mapping information and a transition field for determining a validity of the block map field information;

(a3) generating block map information by recording the mapping information of bad blocks found through an inspection of bad blocks existing in the flash memory and spare blocks to replace the bad blocks found and initially mounting the mapping information on the count field; and

(a4) recording the block map information on any one of the block map pages belonging to the block map page group.

3. The method as claimed in claim 1, wherein the block map page group comprises at least two spare blocks.

4. The method as claimed in claim 1, wherein the flash memory comprises a plurality of chips and each of the steps (a)-(c) is performed for each of the plurality of chips.

5. The method as claimed in claim 1, wherein in step (a), at least a part of the used blocks and the spare blocks are designated as a protected area to block an unauthenticated software from accessing the protected area.

6. The method as claimed in claim 1, wherein in step (b), a valid block map page comprising a latest information-written block page is inspected among the block map pages within the block map page group and

mapping information of the found block map page is resided in the flash memory.

7. The method as claimed in claim 6, wherein when the latest information-written block map page is inspected, an inspection is made based on a value recorded on a transition field of each block map page and the size of a number recorded on the count field.

8. The method as claimed in claim 6, further comprising recovering a read bad block discovered through a predetermined method, where an invalid block map page indicated with a read error is discovered while a valid block map page comprising the latest block map page is inspected among the block map pages belonging to the block map page group.

9. The method as claimed in claim 6, wherein when the flash memory comprises a plurality of chips, each of the steps (a)-(c) is performed for each of the plurality of chips.

10. The method as claimed in claim 1, wherein step (c) comprises:

(c1) locating an unused spare block by a use of the resident mapping information;

(c2) updating the resident mapping information by mapping a number of the unused spare block found through step (c1) with a number of the bad block, as updated mapping information;

(c3) locating an unused block map page of the block map page group as a found unused spare block;

(c4) copying information recorded on the generated bad block on the found unused spare block;

(c5) recording information including the updated mapping information on the found unused block map page; and

(c6) indicating that the found unused block map page is valid.

11. The method as claimed in claim 10, further comprising changing the protected area to be in a readable and writable state when the bad block is generated in the protected area and again changing the protected area to be in a read-only state after completion of step (c6).

12. The method as claimed in either of claims 10, wherein when the flash memory comprises a plurality of chips, each of the steps (a)-(c) is performed for each of the plurality of chips.

13. The method as claimed in claim 1, wherein step (c) comprises:

(c1) determining whether the bad block generated in the step (c) is a read bad block;

(c2) erasing a spare block used temporarily to recover the read bad block;

(c3) correcting an error of the read bad block and then copying the read bad block onto the spare block;

(c4) locating an unused block map page among the block map page groups as a found block map page;

(c5) indicating on the found block map page that the mapping information between the read bad block and the spare block is in recovery from a read error;

(c6) erasing the read bad block as an erased read bad block;

(c7) copying the spare block on the erased read bad block; and

(c8) indicating on the found block map page that the found block map page is invalid.

14. The method as claimed in claim 13, wherein the protected area is changed to be in a readable and writable state when the bad block is generated in the protected area and the protected area is again changed to be in a read-only state after completion of step (c8).

15. An apparatus for managing a bad block in a flash memory constructed with at least one flash memory chip, comprising:

a first flash memory comprising a used area having plural used blocks and a spare area to replace a bad block generated during a use of the flash memory, and comprising within the spare area a block map page group having plural block map pages in which mapping information between the bad block generated during the use and a spare block to replace the bad block, is disposed;

a second memory for loading a flash device manger which defines mapping information of a latest valid block map page existing within the block map page group and a process handling the bad block when generated; and

a central processing unit (CPU) connected to the first flash memory and the second memory, managing a flash operation in the first flash memory of the bad block by reading out a code of the flash device manager.

16. The apparatus as claimed in claim 15, wherein the spare area comprises at least two or more spare blocks.

17. The apparatus as claimed in claim 15, further comprising a flash memory controller connected between the CPU and the first flash memory and comprising two or more buffers storing data for a next operation of the first flash memory while the CPU is doing a current flash operation.

18. The apparatus as claimed in claim 17, wherein the flash memory controller establishes a predetermined area of the first flash memory as a protected area and allows a flash operation in the protected area only for a flash application authenticated by the flash device manager and does not allow the flash operation in the protected area for a flash application not authenticated by the flash device manager.

19. The apparatus as claimed in claim 15, wherein when the first flash memory comprises plural flash memory chips, and a corresponding used area, a corresponding spare area and a corresponding block map page group

within the corresponding spare area of the first flash memory exist in each of the plural flash memory chips.